

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of processing a network connection in a computer system, comprising:
 - establishing the network connection by a first network protocol stack;
 - determining whether to offload the network connection from the first network protocol stack to a second network protocol stack;
 - transferring the network connection from the first network protocol stack to the second network protocol stack when it is determined to offload the network connection from the first network protocol stack to the second network protocol stack, and
 - determining whether to accept or refuse the transfer of the network connection at the second network protocol stack based at least in part on a state of the second network protocol stack and a nature of the network connection, wherein determining whether to accept a determination to refuse the transfer further comprises refusing the transfer of the network connection based at least in part on the state of the second network protocol stack and the nature of the network connection, and wherein a determination to accept the transfer of the network connection at the second network protocol stack further comprises accepting the transfer if the network connection exceeds the capability of the second network protocol network stack.
2. (Original) The method as recited in claim 1, further comprising:
 - sharing state information associated with the network connection between the first network protocol stack and the second network protocol stack.
3. (Original) The method as recited in claim 1, wherein determining whether to offload the network connection is performed by an operating system kernel of the computer system.
4. (Original) The method as recited in claim 3, wherein determining whether to offload the network connection is performed by a socket layer of the operating system kernel.

5. (Original) The method as recited in claim 1, wherein determining whether to offload the network connection is performed by the first network protocol stack.
6. (Original) The method as recited in claim 1, wherein the first network protocol stack is implemented in software and the second network protocol stack is implemented in hardware.
7. (Original) The method as recited in claim 1, wherein the first network protocol stack is implemented in at least one of software and hardware, and the second network protocol stack is implemented in at least one of software and hardware.
8. (Original) The method as recited in claim 6, wherein the hardware is a TOE capable NIC.
9. (Original) The method as recited in claim 1, wherein the second network protocol stack is capable of determining whether to offload the network connection back to the first network protocol stack.
10. (Original) The method as recited in claim 9, further comprising:
receiving an indicator from the second network protocol stack or a driver associated with the second network protocol stack, the indicator indicating a request to transfer the network connection back to the first network protocol stack.
11. (Original) The method as recited in claim 10, further comprising:
obtaining state information for the network connection from the second network protocol stack or the driver associated with the second network protocol stack when the indicator is received;
and
handling the network connection by the first network protocol stack using the obtained state information.

12. (Original) The method as recited in claim 11, wherein obtaining state information is performed by a TCP layer of the first network protocol stack.
13. (Original) The method as recited in claim 11, further comprising:
obtaining at least one of unsent and undelivered data by the first network protocol stack from the second network protocol stack or a driver associated with the second network protocol stack, thereby enabling the first network protocol stack to process the unsent or undelivered data.
14. (Original) The method as recited in claim 9, wherein the first network protocol stack is implemented in software and the second network protocol stack is implemented in hardware.
15. (Original) The method as recited in claim 14, wherein the hardware is a TOE capable NIC.
16. (Original) The method as recited in claim 15, wherein the network connection is a TCP connection.
17. (Original) The method as recited in claim 9, further comprising:
handling the network connection by the first network protocol stack when the network connection is offloaded back to the first network protocol stack from the second network protocol stack.
18. (Original) The method as recited in claim 9, further comprising:
handling the network connection by the first network protocol stack until it is determined to offload the network connection to the second network protocol stack.

19. (Original) The method as recited in claim 1, further comprising:
handling the network connection by the first network protocol stack until it is determined to offload the network connection to the second network protocol stack.
20. (Original) The method as recited in claim 1, further comprising:
providing state information associated with the first network protocol stack to the second network protocol stack when it is determined to offload the network connection from the first network protocol stack to the second network protocol stack.
21. (Original) The method as recited in claim 1, further comprising:
establishing a mapping between a first set of state information for the network connection maintained by or associated with the first network protocol stack and a second set of state information for the network connection maintained by or associated with the second network protocol stack.
22. (Original) The method as recited in claim 1, wherein transferring the network connection from the first network protocol stack to the second network protocol stack comprises:
exchanging state information for the network connection between the first network protocol stack and the second network protocol stack.
23. (Original) The method as recited in claim 22, wherein exchanging state information comprises:
exchanging a first identifier for the network connection maintained by the first network protocol stack with a second identifier for the network connection maintained by the second network protocol stack.

24. (Original) The method as recited in claim 22, wherein the state information comprises IP addresses and ports for a client and server of the network connection, and at least one of send and receive sequence numbers of one or more packets for the network connection.

25. (Original) The method as recited in claim 24, wherein the state information further comprises:

a round trip estimate.

26. (Original) The method as recited in claim 25, wherein the state information further comprises:

a congestion window and slow start information.

27. (Original) The method as recited in claim 1, wherein transferring the network connection from the first network protocol stack to the second network protocol stack comprises:

initiating the transfer of the network connection by a socket layer of the computer system.

28. (Original) The method as recited in claim 1, wherein upon transferring the network connection from the first network protocol stack to the second network protocol stack, the method further comprising:

at least one of sending one or more packets by the second network protocol stack to the socket layer and receiving one or more packets by the second network protocol stack from the socket layer.

29. (Currently Amended) A method of processing a network connection in a computer system, comprising:

establishing the network connection by an operating system of the computer system;

determining whether to offload the network connection from the operating system to a network interface card;

transferring the network connection from the operating system to the network interface card when it is determined to offload the network connection from the operating system to the network interface card, and

determining whether to accept or refuse the transfer of the network connection at the network interface card based at least in part on a state of the network interface card and a nature of the network connection, wherein a determination to refuse determining whether to accept the transfer further comprises refusing the transfer of the network connection based at least in part on the state of the network interface card and the nature of the network connection, and wherein a determination to accept the transfer of the network connection at the second network protocol stack further comprises accepting the transfer if the network connection exceeds the capability of the second network protocol network stack.

30. (Currently Amended) An apparatus for processing a network connection in a computer system, comprising:

means for establishing the network connection by a first network protocol stack;

means for determining whether to offload the network connection from the first network protocol stack to a second network protocol stack;

means for transferring the network connection from the first network protocol stack to the second network protocol stack when it is determined to offload the network connection from the first network protocol stack to the second network protocol stack, and

means for determining whether to accept or refuse the transfer of the network connection at the second network protocol stack based at least in part on a state of the second network protocol stack and a nature of the network connection, wherein a determination to refuse determining whether to accept the transfer further comprises refusing the transfer of the network connection based at least in part on the state of the second network protocol stack and the nature of the network connection, and wherein a determination to accept the transfer of the network connection at the second network protocol stack further comprises accepting the transfer if the network connection exceeds the capability of the second network protocol network stack.

31. (Currently Amended) A computer-readable medium storing thereon computer-readable instructions for processing a network connection in a computer system, comprising:
- instructions for establishing the network connection by a first network protocol stack;
- instructions for determining whether to offload the network connection from the first network protocol stack to a second network protocol stack;
- instructions for transferring the network connection from the first network protocol stack to the second network protocol stack when it is determined to offload the network connection from the first network protocol stack to the second network protocol stack, and
- instructions for determining whether to accept or refuse the transfer of the network connection at the second network protocol stack based at least in part on a state of the second network protocol stack and a nature of the network connection, wherein a determination to refuse determining whether to accept the transfer further comprises refusing the transfer of the network connection based at least in part on the state of the second network protocol stack and the nature of the network connection, and wherein a determination to accept the transfer of the network connection at the second network protocol stack further comprises accepting the transfer if the network connection exceeds the capability of the second network protocol network stack.
32. (Currently Amended) A network device adapted for processing a network connection, comprising:
- an operating system including a first network protocol stack;
- a second network protocol stack coupled to the first network protocol stack, the operating system being configured for determining whether to offload a network connection to the second network protocol stack and transferring the network connection from the first network protocol stack to the second network protocol stack when it determines that it will offload the network connection to the second network protocol stack; and
- a control component being configured for determining whether to accept or refuse the transfer of the network connection at the second network protocol stack based at least in part on a

state of the second network protocol stack and a nature of the network connection, wherein a determination to refuse determining whether to accept the transfer further comprises refusing the transfer of the network connection based at least in part on the state of the second network protocol stack and the nature of the network connection, and wherein a determination to accept the transfer of the network connection at the second network protocol stack further comprises accepting the transfer if the network connection exceeds the capability of the second network protocol network stack.

33. (Original) The network device as recited in claim 32, wherein the first network protocol stack is a TCP/IP stack and the second network protocol stack is a TCP/IP stack.

34. (Previously presented) The network device as recited in claim 32, wherein the first network protocol stack is implemented in software and the second network protocol stack and control component are implemented in hardware.

35. (Original) The network device as recited in claim 34, wherein the hardware is a TOE capable NEC.

36. (Original) The network device as recited in claim 32, wherein the second network protocol stack is capable of determining whether to offload the network connection back to the first network protocol stack.

37. (Original) The network device as recited in claim 36, wherein the second network protocol stack or a driver associated with the second network protocol stack sends an indicator when it requests to transfer the network connection back to the first network protocol stack.

38. (Original) The network device as recited in claim 32, wherein the first network protocol stack is adapted for obtaining state information for the network connection from the second network

protocol stack or a driver associated with the second network protocol stack when an indicator is received, thereby enabling the first network protocol stack to handle the network connection using the obtained state information.

39. (Original) The network device as recited in claim 38, wherein a TCP layer of the first network protocol stack is adapted for obtaining the state information for the network connection from the second network protocol stack or the driver associated with the second network protocol stack.

40. (Original) The network device as recited in claim 38, wherein the first network protocol stack is further adapted for obtaining at least one of unsent and undelivered data from the second network protocol stack, thereby enabling the first network protocol stack to process the unsent or undelivered data.

41. (Original) The network device as recited in claim 38, wherein the indicator is received from the second network protocol stack or a driver associated with the second network protocol stack.

42. (Previously presented) The network device as recited in claim 36, wherein the first network protocol stack is implemented in software and the second network protocol stack and control component are implemented in hardware.

43. (Original) The network device as recited, in claim 42, wherein the hardware is a TOE capable NIC.

44. (Original) The network device as recited in claim 43, wherein the network connection is a TCP connection.

45. (Original) The network device as recited in claim 36, wherein the first network protocol stack is capable of handling the network connection when the network connection is offloaded back to the first network protocol stack from the second network protocol stack.
46. (Original) The network device as recited in claim 45, wherein the first network protocol stack handles the network connection until it is determined by the operating system to offload the network connection to the second network protocol stack.
47. (Original) The network device as recited in claim 32, wherein the first network protocol stack handles the network connection until it is determined by the operating system to offload the network connection to the second network protocol stack.
48. (Original) The network device as recited in claim 32, wherein the operating system is configured to establish a mapping between a first set of state information for the network connection associated with the first network protocol stack and a second set of state information for the network connection associated with the second network protocol stack.
49. (Original) The network as recited in claim 32, wherein the operating system is configured for providing state information for the network connection to the second network protocol stack when the first network protocol stack decides to offload the network connection from the first network protocol stack to the second network protocol stack.
50. (Original) The network device as recited in claim 32, wherein the operating system is configured to at least one of provide state information associated with the first network protocol stack and obtain state information associated with the second network protocol stack.

51. (Original) The network device as recited in claim 50, wherein the state information comprises an identifier for the network connection.

52. (Original) The network device as recited in claim 50, wherein the state information comprises EP addresses and ports for a client and server of the connection, and at least one of send and receive sequence numbers of one or more packets for the connection.

53. (Original) The network device as recited in claim 52, wherein the state information further comprises:

a round trip estimate.

54. (Original) The network device as recited in claim 53, wherein the state information further comprises:

a congestion window and slow start information.

55. (Original) The network device as recited in claim 32, wherein the transfer of the network connection from the first network protocol stack to the second network protocol stack is initiated by a socket layer of the first network protocol stack.

56. (Original) The network device as recited in claim 32, wherein upon transferring the network connection from the first network protocol stack to the second network protocol stack, the second network protocol stack is in communication with a socket layer of the first network protocol stack, thereby enabling data to be sent by the second network protocol stack to the socket layer and enabling data to be received by the second network protocol stack from the socket layer.